ECE 376: Binary Clock

Spring 2011 - C

Purpose:

- Learn how to compile and download a program onto your PIC board
- Get some familiarity with loops and timing in C

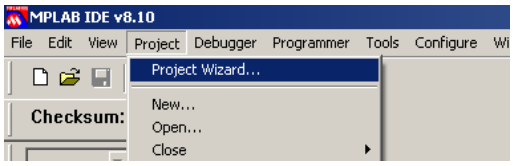
Requirements:

- Display the time on the LEDs on your PIC board as
  - PORTD = seconds
  - PORTC = minutes
  - PORTB = hours

Procedure:

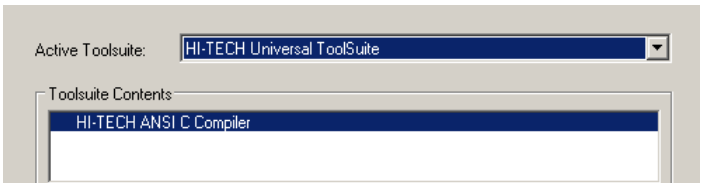
Step 1) Create a directory on your z: drive: z:\ECE376\Lab1. On-line, open the zip file under ECE 376 for Lab #1. Select a program which works and has some features you'll use, such as Clock.zip. Copy all of the files in this zip file into this directory.

Step 2: Compiling) Start MPLAB. Go to Project Wizard

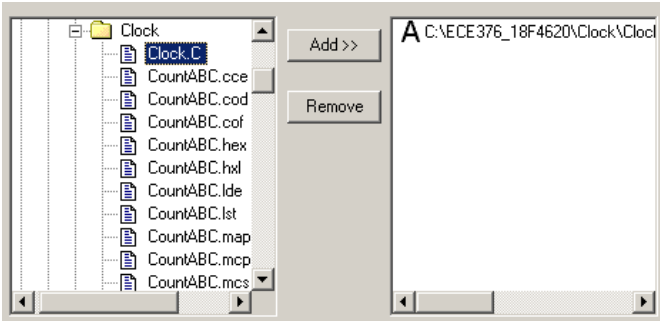


Select PIC18F4620

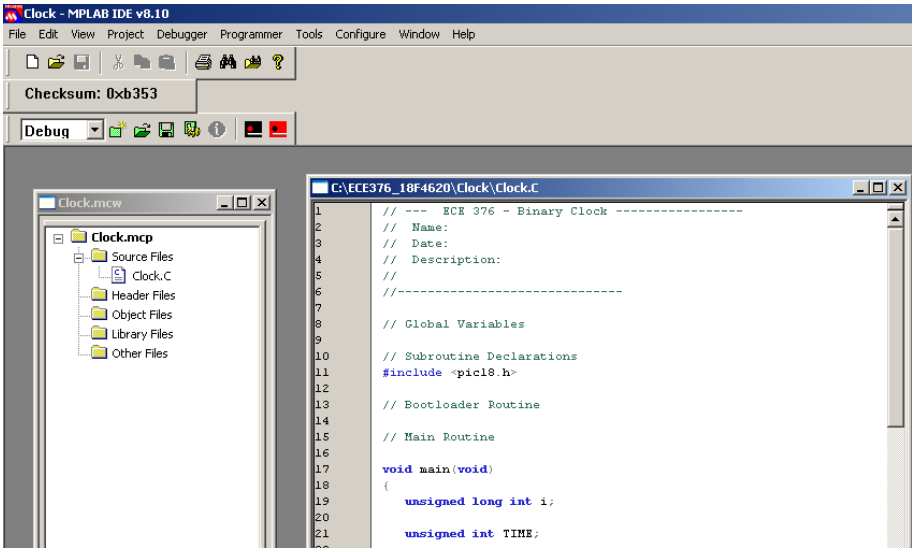
- Universal Tool Suite (the C compiler we're using)



- z:ECE376/Lab1/Lab1.mcp for your project
- Clock.C for the program to include.

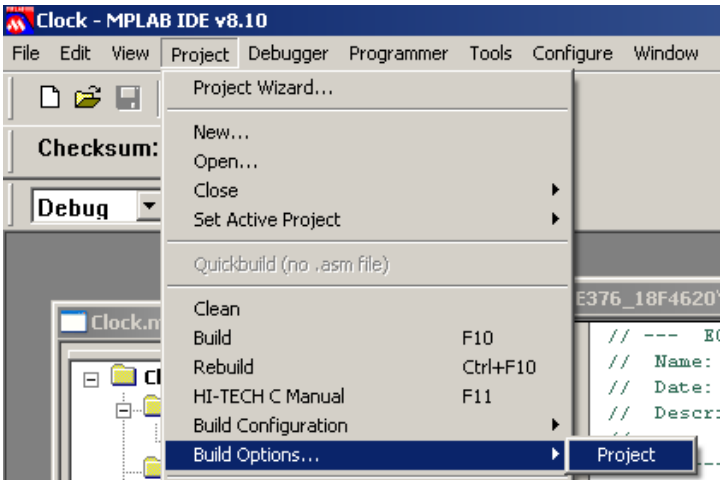


What you should see is the following. The left window is the project you're going to compile. It contains one C program, CLOCK.C. If you close the C program, double clock on Clock.C, you can make sure that the file you're editing (on the right) is the file you're compiling. It's pretty annoying if you're editing the wrong file - the compiler won't tell you if you are.

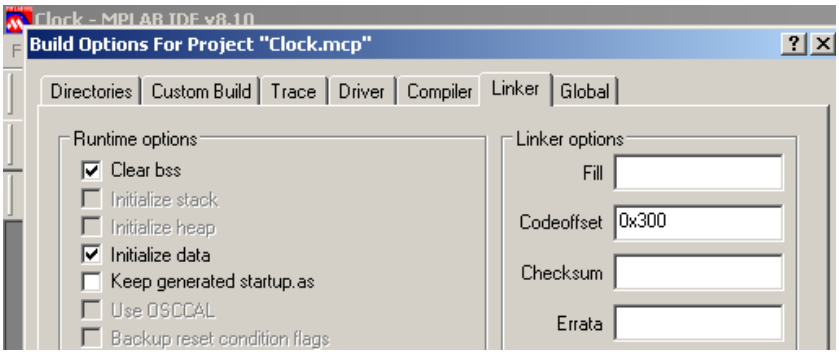


Important: Your code must be offset by 0x300. The low 0x300 spots in memory are the boot loader. If you forget to offset your code and you call a subroutine, your program crashes.

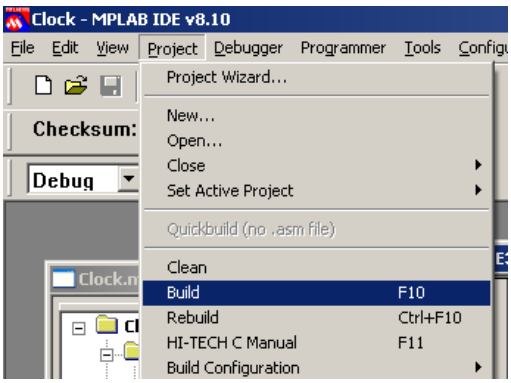
To offset your code, go to Project Build Options



Click on Linker and type in 0x300 into the code offset.

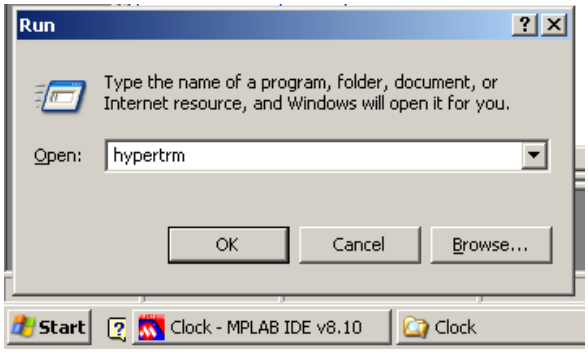


Compile this program (Project Build or control F10).



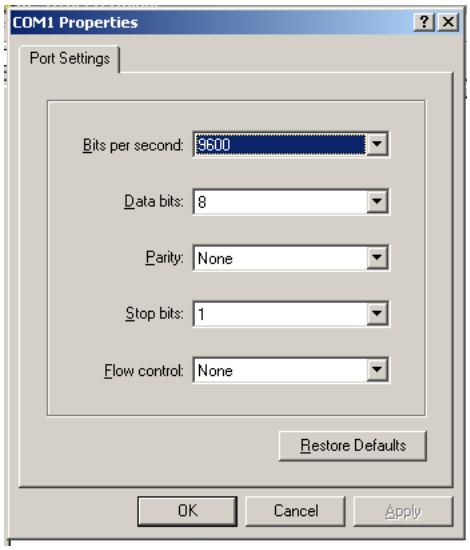
Step 3: Dowlonading a hex file) Start hyperterminal

- in Windows, click on the Start menu, Run, hypertrm



Set up hyperterminal for

- 8 data bits, no parity, 1 stop bit, 9600 baud, no handshaking.



Connect your PIC board to the serial port and turn on hyperterminal. When you press the reset button, you should get the message '321'.

Press reset and hit the return key on the PC before you get to zero. You'll see a prompt.

In hyperterminal, select Transfer Text File and send Lab1.hex. You should see your program running.

Step 4) Bottom Up Programming)

4a) Modify this program so that it counts one tic per second. hint: An oscilloscope helps to get the timing right.

4b) Change the code so that seconds count up to 9 and then goes to zero. When it gets to 9, the minutes should increment. (You might change the wait loop to speed it up a little while you're debugging your code.)

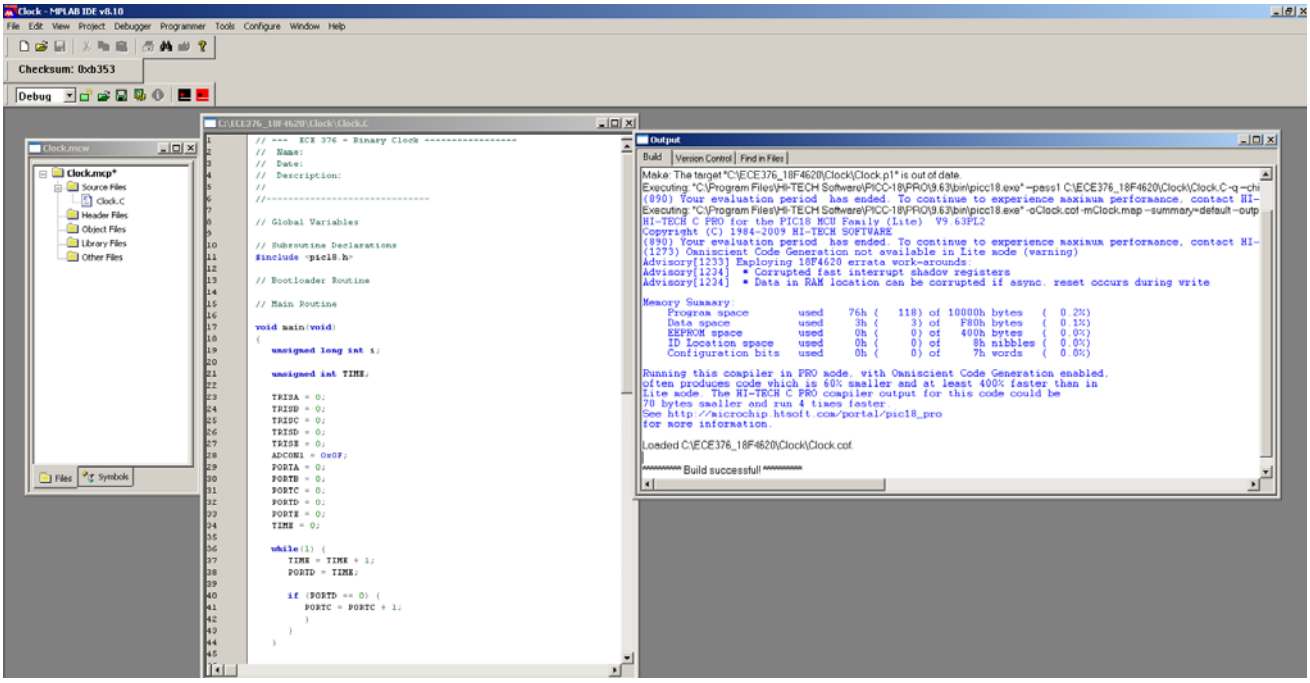
4c) Modify this program so that the minutes count to 9, they go to zero and hours increment by one.

4d) Modify this program so that hours count from 1 to 12 and then go back to 1.

Step 5: Modify your code so that seconds and minutes actually count to 59 (rather than 9).

Writeup: Include

- A flow chart for your final code
- The C listing for your final code
- The size of your final code. (View Program Listing after you compile and you can see where your program is located in memory. It should be at 0x300+)


Clock - MPLAB IDE v8.10
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xb353
Debug
Clock.mcw
Clock.mcp*
Source Files
Clock.C
Header Files
Object Files
Library Files
Other Files
Files
Symbols
C:\ECE376_18F4620\Clock\Clock.C
// === ECE 376 - Binary Clock ==========
// Name:
// Date:
// Description:
//
//------------------------------
// Global Variables
// Subroutine Declarations
#include <pic18.h>
// Bootloader Routine
// Main Routine
void main(void)
{
unsigned long int i;
unsigned int TIME;
TRISA = 0;
TRISB = 0;
TRISC = 0;
TRISD = 0;
TRISE = 0;
ADCON1 = 0x0F;
PORTA = 0;
PORTB = 0;
PORTC = 0;
PORTD = 0;
PORTE = 0;
TIME = 0;
while(1) {
TIME = TIME + 1;
PORTD = TIME;
if (PORTD == 0) {
PORTC = PORTC + 1;
}
}
Output
Build
Version Control
Find in Files
Make: The target "C:\ECE376_18F4620\Clock\Clock.p1" is out of date.
Executing: "C:\Program Files\HI-TECH Software\PICC-18\PRO\9.63\bin\picc18.exe" --pass1 C:\ECE376_18F4620\Clock\Clock.C -q --chi
(890) Your evaluation period has ended. To continue to experience maximum performance, contact HI-
Executing: "C:\Program Files\HI-TECH Software\PICC-18\PRO\9.63\bin\picc18.exe" -oClock.cof -mClock.map --summary=default --outp
HI-TECH C PRO for the PIC18 MCU Family (Lite) V9.63PL2
Copyright (C) 1984-2009 HI-TECH SOFTWARE
(890) Your evaluation period has ended. To continue to experience maximum performance, contact HI-
(1273) Omniscient Code Generation not available in Lite mode (warning)
Advisory[1233] Enploying 18F4620 errata work-arounds:
Advisory[1234] * Corrupted fast interrupt shadow registers
Advisory[1234] * Data in RAM location can be corrupted if async. reset occurs during write
Memory Summary:
Program space used 76h (118) of 10000h bytes (0.2%)
Data space used 3h (3) of F80h bytes (0.1%)
EEPROM space used 0h (0) of 400h bytes (0.0%)
ID Location space used 0h (0) of 8h nibbles (0.0%)
Configuration bits used 0h (0) of 7h words (0.0%)
Running this compiler in PRO mode, with Omniscient Code Generation enabled,
often produces code which is 60% smaller and at least 400% faster than in
Lite mode. The HI-TECH C PRO compiler output for this code could be
70 bytes smaller and run 4 times faster.
See http://microchip.htsoft.com/portal/pic18_pro
for more information.
Loaded C:\ECE376_18F4620\Clock\Clock.cof.
********** Build successful! **********